

Figura 10.24 Unità Logica Booleana.

10.6 Matrici Logiche Programmabili (PLA)

L'applicazione più versatile e flessibile dei circuiti combinatori è quella dei *Componenti Logici Programmabili (Programmable Logic Device, PLD)*, che sono essenzialmente dei circuiti logici a struttura regolare integrati in un unico chip, i quali possono essere modificati nelle connessioni interne dall'utente in modo da realizzare una qualsiasi espressione logica relativamente complessa basata su espressioni del tipo di somme di prodotti. Discuteremo più dettagliatamente questi componenti nel Capitolo 12, dopo aver introdotto i circuiti combinatori, che sono inseriti nelle versioni più complesse di questi componenti. Nell'ambito dei circuiti combinatori, la versione più elementare di questi circuiti è quella delle *Matrici Logiche Programmabili (Programmable Logic Arrays, PLA)* che è basata su una struttura regolare di celle AND connesse in una matrice, detta appunto matrice (o piano) AND, le cui uscite sono connesse a una seconda matrice di porte OR, detta matrice (o piano) OR, come è indicato schematicamente nella Figura 10.47.

La matrice di P porte AND, ognuna delle quali presenta N ingressi, può fornire p termini di prodotto ognuno con N variabili; la matrice di M porte OR può fornire M termini di somma tra i P prodotti, per cui in definitiva si possono ottenere M espressioni logiche ognuna con P termini di somma e N prodotti per ogni termine. Le specifiche funzioni logiche che si desidera realizzare vengono implementate nel circuito o

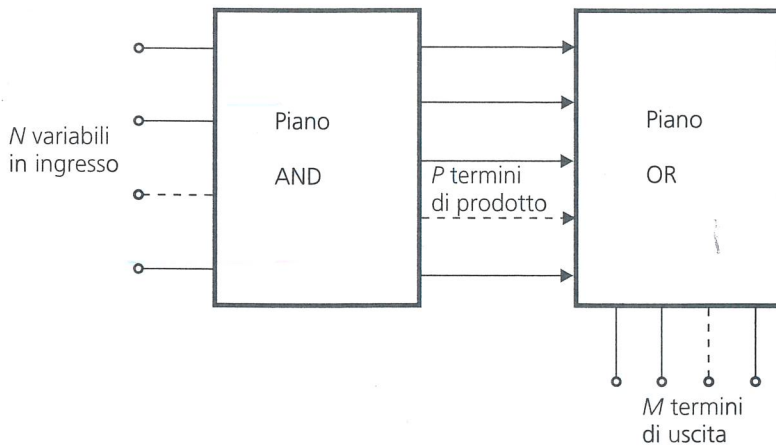


Figura 10.25 Schema a blocchi di un PLA.

fornisca in di opportunità del abili (AND, :ombinazio- ssunti dalla ca funzione e dati. Que- si valori as- ottenibili in

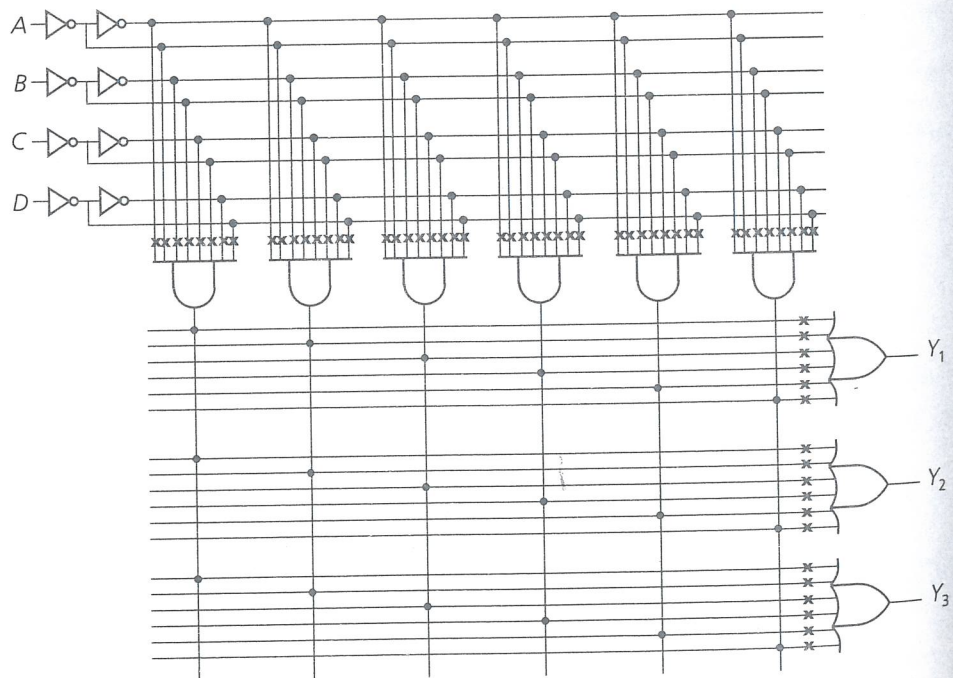
P_4
0
1
0
1
0
1

eliminando le connessioni non volute rispettivamente nel piano AND e in quello OR, o inserendo connessioni elettriche tra collegamenti, mediante speciali dispositivi detti rispettivamente fusibili o antifusibili; le operazioni per la programmazione mediante questi collegamenti da inserire o eliminare vengono effettuate dall'utente direttamente sull'integrato, mediante applicazione di opportune tensioni ai collegamenti che si debbono inserire o eliminare. Per esempio la funzione di fusibile è realizzata utilizzando per le linee di interconnessione metalli opportuni con una sezione ridotta nel percorso della corrente; applicando una corrente determinata superiore a quella di esercizio il collegamento si interrompe perché il metallo evapora. La funzione di antifusibile viene realizzata sovrapponendo due linee di metallo separate da un sottile strato di dielettrico; se si applica una tensione opportuna tra le due linee il dielettrico viene forato e si crea un contatto con resistenza dell'ordine delle centinaia di ohm.

Nella Figura 10.26 è riportato lo schema di una piccola PLA di dimensioni $N = 4$, $P = 6$, $M = 3$. In questo schema le x indicano le connessioni possibili delle matrici che possono essere conservate o eliminate; ogni ingresso viene inviato a stadi invertitori che forniscono sia le variabili dirette che quelle negate ai possibili ingressi delle porte AND. Le realizzazioni circuitali delle matrici AND e OR delle PLA sono analoghe a quelle viste rispettivamente per i circuiti decodificatori e codificatori, e possono essere realizzate sia in tecnologia bipolare che MOS.

Per esempio, una versione di una PLA in tecnologia bipolare che utilizza diodi sia nel piano AND che in quello OR è riportata nella Figura 10.27; le possibili connessioni sono realizzate mediante diodi connessi in serie con fusibili che possono essere aperti applicando una tensione opportuna alle specifiche righe e colonne della matrice. Le uscite della matrice superiore sono inviate a invertitori, per cui si ottiene una funzione logica complessiva NAND; anche la seconda matrice, che realizza la funzione AND tra righe e colonne, ha le uscite connesse a stadi invertitori (che effettuano anche la funzione di stadi buffer), per cui la funzione complessiva effettuata dalle due matrici è una funzione NAND-NAND tra le variabili, che equivale a una funzione AND-OR.

Figura 10.26 Schema logico di una PLA con 4 ingressi, 3 uscite e 6 termini di prodotto.



A →
B →
C →
D →

Una ve
portata
di deco

A₁ —
A₂ —
A₃ —
A₄ —

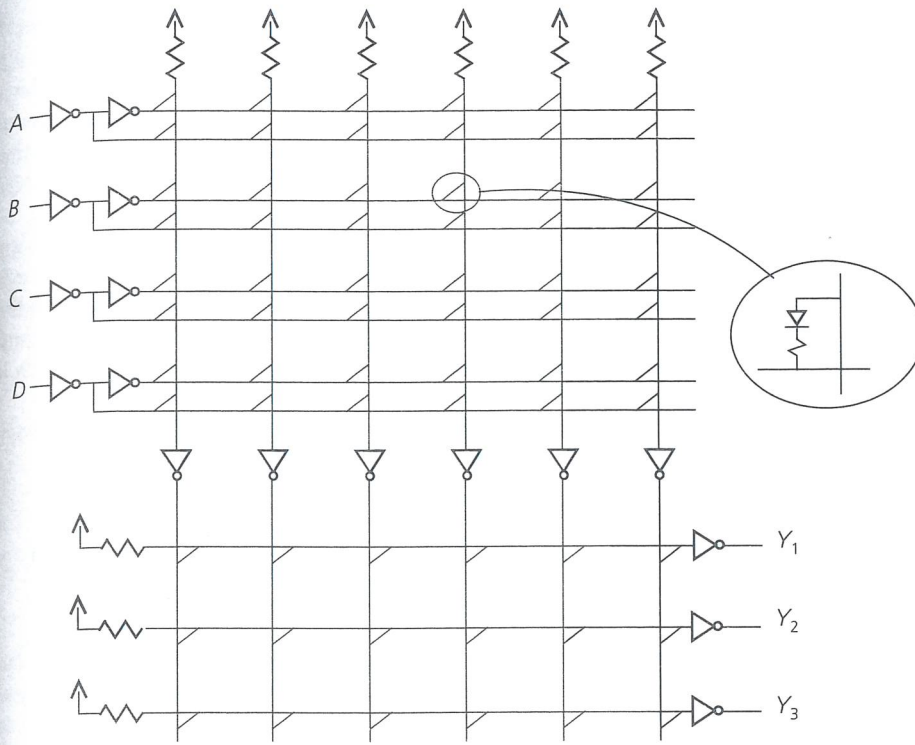


Figura 10.27 Matrici a diodi per PLA a 4 ingressi e 3 uscite.

Una versione in tecnologia NMOS delle matrici dei piani AND e OR di una PLA è riportata nella Figura 10.28. In questo caso, analogamente a quanto visto per le matrici di decodifica a MOS, conviene adottare porte elementari NOR, dove le linee degli in-

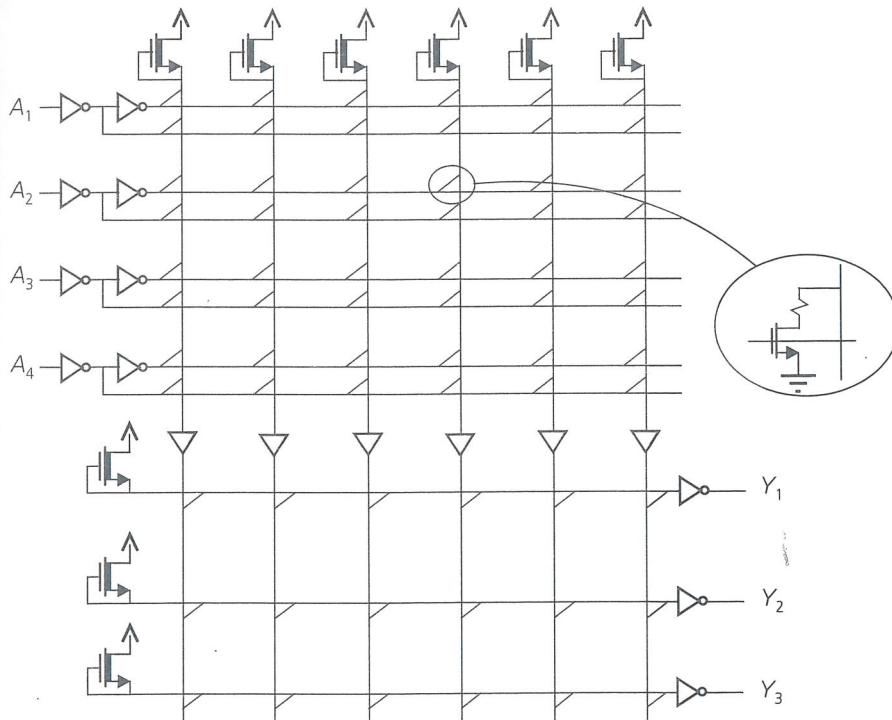


Figura 10.28 Matrici a MOS per PLA a 4 ingressi e 3 uscite.

lo OR,
vi detti
ediate
amente
si deb-
zzando
rcorso
zizio il
le vie-
dielet-
orato e

V = 4,
matrici
nverti-
si delle
analo-
ossono

odi sia
nessio-
essere
atrice.
a fun-
zione
no an-
lle due
nzione

— Y₁
— Y₂
— Y₃

gressi pilotano le diverse gate dei MOS e le uscite connettono i drain in parallelo. Il collegamento tra i due piani viene effettuato con stadi buffer non invertenti, per cui la prima matrice, scegliendo opportunamente le variabili nel modo già visto per il decodificatore NMOS, equivale alla funzione AND tra gli ingressi e le uscite. Il secondo piano realizza ancora una funzione NOR, ma le uscite sono applicate a stadi invertitori e realizzano la funzione OR; la funzione logica complessiva è perciò quella AND-OR richiesta.

Gli schemi a matrici delle figure precedenti giustificano la rappresentazione compatta usualmente utilizzata per le PLA, riportata nella Figura 10.29 (che corrisponde allo schema esteso di Figura 10.26). In questo caso il simbolo della porta AND indica la funzione complessiva realizzata tra gli ingressi connessi alla singola linea di ingresso a quella porta; lo stesso vale per il simbolo della porta OR per quanto riguarda le uscite AND connesse al singolo ingresso della porta OR.

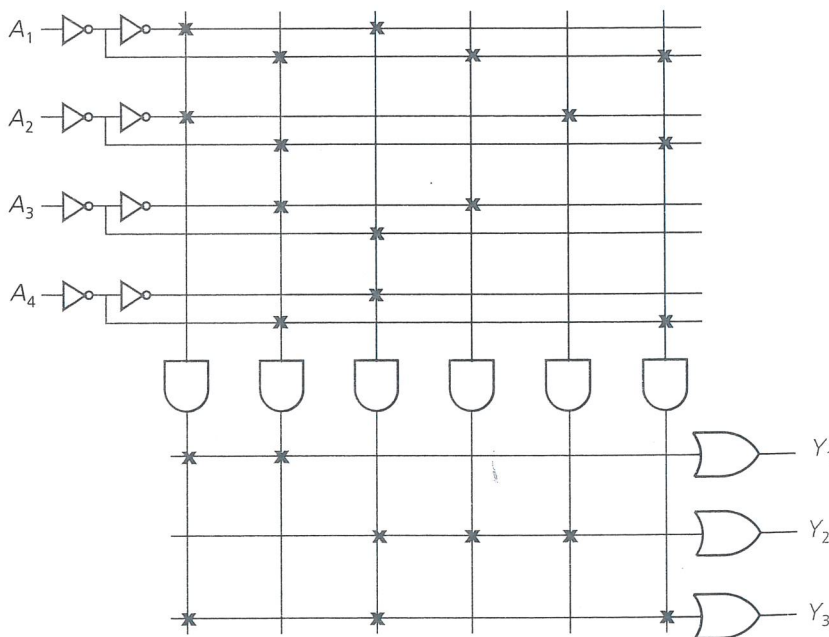
Nel caso della Figura 10.29, per esemplificare l'interpretazione dello schema logico, le connessioni nei piani AND e OR realizzano le seguenti funzioni logiche:

$$\begin{aligned} Y_1 &= A_1 \cdot A_2 + \bar{A}_1 \cdot \bar{A}_2 \cdot A_3 \cdot \bar{A}_4 \\ Y_2 &= A_1 \cdot \bar{A}_3 \cdot A_4 + \bar{A}_1 \cdot A_3 + A_2 \\ Y_3 &= A_1 \cdot A_2 + A_1 \cdot \bar{A}_3 \cdot A_4 + \bar{A}_1 \cdot \bar{A}_2 \cdot \bar{A}_4 \end{aligned} \quad (10.12)$$

Poiché nei PLA non si utilizzano tutte le combinazioni possibili degli ingressi nei due piani, ma solo quelle richieste per implementare le funzioni logiche desiderate, si può avere una forte dispersione dei dispositivi (MOS o diodi) nelle rispettive matrici, con conseguente sottoutilizzazione dell'area a disposizione. Per esempio nella matrice AND del PLA della Figura 10.12 solo 15 delle 48 possibili intersezioni della matrice sono utilizzate (ossia prevedono un dispositivo per la funzione logica da realizzare).

È possibile compattare la matrice effettuando operazioni di taglio lungo le linee degli ingressi in modo da alimentare una parte della linea con la variabile vera e il resto della linea con quella negata (ricordiamo che ogni porta prevede l'uso di una variabile o del suo negato e non di entrambi), e di riorganizzazione dell'ordine delle linee a val-

Figura 10.29 Schema logico compatto della PLA che realizza le funzioni (10.12).



le del
ne del
gura l
arriva
di part
I co
digital
nologi
media
mento

Eserc

- 10.1 Pe
rip
01
- 10.2 Va
Fi
- 10.3 Pe
ter
pe
pa
co
- 10.4 Re
- 10.5 Re
ch
luz
me
del

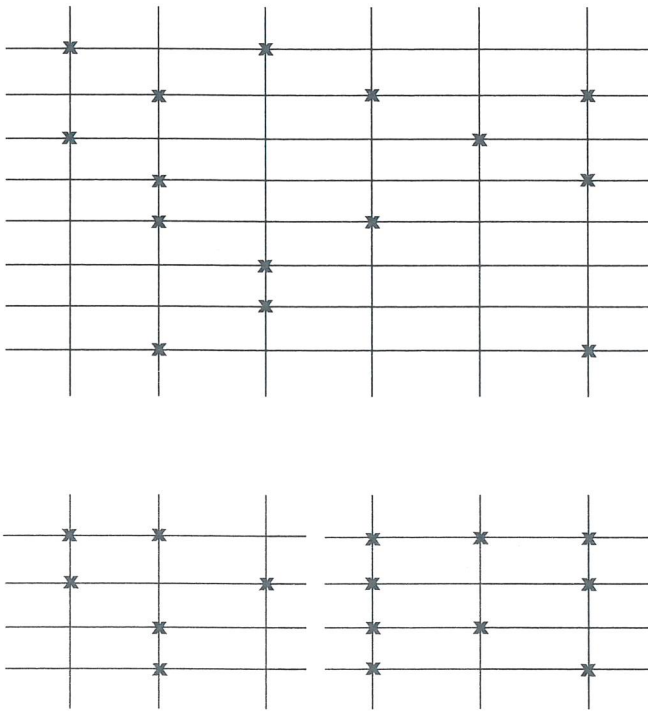


Figura 10.30 Taglio e riorganizzazione della matrice AND del PLA di Figura 10.29.

le del taglio; come esempio si è effettuata questa operazione di taglio e riorganizzazione della sequenza di linee per lo schema compatto della matrice AND del PLA di Figura 10.29, giungendo alla matrice più compatta di Figura 10.30. In quest'ultima, si arriva a un grado di utilizzazione della matrice del 62% rispetto al 31% della matrice di partenza.

I componenti PLA hanno avuto un largo sviluppo nella realizzazione dei sistemi digitali per la loro flessibilità di impiego; ciò vale in particolare per le versioni con tecnologia MOS che permettono la riprogrammazione della funzione logica effettuata mediante operazioni direttamente realizzate dall'utente; ritorneremo su questo argomento trattando dei PLD nel Capitolo 12.

Esercizi di riepilogo

- 10.1 Per un sommatore a propagazione del riporto a 8 bit, assumendo un ritardo di propagazione del riporto $t_P = 1$ ns per ogni full adder, valutare il tempo necessario a ottenere la somma di a) $01111111 + 00000001$ e b) $01111111 + 01000000$.
- 10.2 Valutare il numero di transistori necessari a realizzare le diverse configurazioni riportate nelle Figure 10.5, 10.6 in logica CMOS.
- 10.3 Per il decodificatore NMOS della Figura 10.14, alimentato con $V_{DD} = 5$ V, determinare: a) il tempo di propagazione t_{PLH} della linea indirizzata; b) la potenza dissipata nel decodificatore per qualsiasi combinazione dei bit di ingresso. Si assumano gli NMOS di carico con i seguenti parametri: $k'_N = 50 \mu\text{A}/\text{V}^2$, $W_N = 2 \mu\text{m}$, $L_N = 4 \mu\text{m}$, $V_{TD} = -3$ V, e quelli ad arricchimento con: $k'_N = 50 \mu\text{A}/\text{V}^2$, $W_N = 4 \mu\text{m}$, $L_N = 2 \mu\text{m}$, $V_T = 0.8$ V.
- 10.4 Realizzare uno schema circuitale di decodificatore 8/3 in tecnologia Pseudo-NMOS.
- 10.5 Realizzare un multiplexer con 8 ingressi utilizzando due multiplexer da 4 ingressi, e porte logiche NOR. Valutare il tempo di propagazione complessivo e l'occupazione di area di questa soluzione rispetto a quella con un unico multiplexer a 8 ingressi per una tecnologia CMOS, assumendo che i tempi di propagazione di ciascuna porta siano proporzionali al numero di ingressi della porta stessa.

Dispositivi Logici Programmabili 12

12.1 Introduzione

Il progetto di sistemi digitali, fino a non molti anni fa, era basato sull'utilizzo di circuiti logici standard a bassa e a media scala di integrazione, della serie 54/74. Questi circuiti integrati furono inizialmente prodotti in tecnologia TTL standard e in seguito in tecnologia TTL Schottky e Schottky Low-Power. Attualmente sono disponibili anche in versione CMOS. Nei circuiti integrati commercialmente disponibili, le lettere nella sigla consentono di identificare la tecnologia con la quale il circuito è stato realizzato. A titolo di esempio, un circuito integrato caratterizzato dalla sigla 74ACT02 include quattro porte NOR a due ingressi, realizzate in tecnologia CMOS "avanzata", con livelli logici TTL compatibili. I due prefissi "74" e "54" vengono utilizzati per distinguere rispettivamente le versioni commerciali e militari; queste ultime possono operare in un campo di temperature e di tensioni di alimentazioni più esteso.

Il progetto di un sistema digitale basato su circuiti logici standard prevede fondamentalmente i tre passi seguenti: a) definizione delle specifiche; b) descrizione del sistema mediante interconnessione di circuiti logici standard; c) assemblaggio dei circuiti integrati su di un circuito stampato.

Questo approccio, al crescere della complessità del sistema da progettare, richiede un numero sempre maggiore di circuiti SSI o MSI, con aumenti di costo, potenza dissipata e ritardi di propagazione oltre che di ingombro. Un sistema costituito da dispositivi 54/74 può inoltre essere facilmente "clonato", visto che ogni circuito integrato realizza una funzione logica ben nota.

Per sopperire a queste limitazioni sono stati introdotti i dispositivi logici programmabili (PLD), che vengono impiegati sempre più frequentemente al posto dei circuiti integrati TTL. Un PLD è un circuito a elevata scala di integrazione che può essere opportunamente programmato o personalizzato dall'utente finale, in modo da realizzare una specifica funzione.

I vantaggi legati all'utilizzo di PLD sono molteplici. Un singolo PLD può sostituire numerosi circuiti a bassa o media scala di integrazione, con significativi miglioramenti in termini di area occupata sul circuito stampato, di affidabilità e di costi. Le interconnessioni a livello di circuito stampato vengono sostituite da collegamenti all'interno di un singolo circuito integrato. La riduzione delle capacità parassite consente di migliorare sia i tempi di propagazione che la potenza dissipata. Un sistema basato su PLD è molto più flessibile rispetto a uno realizzato con componenti logici discreti. Molti PLD sono infatti riprogrammabili elettricamente anche dopo essere stati collegati su di un circuito stampato. Ciò consente di modificare la funzionalità di un sistema digitale senza dover aggiungere o rimuovere componenti.

Il progetto di un sistema digitale basato su PLD si basa sull'utilizzo di opportuni sistemi di sviluppo che consentono di ottenere in maniera quasi del tutto automatica la mappa di programmazione del PLD a partire da una descrizione ad alto livello. Gli elementi primitivi di cui si dispone vanno dalle semplici porte logiche elementari fino a funzioni più complesse come decodificatori, addizionatori, contatori, registri a scorrimento ecc. Molto spesso i nomi di questi blocchi logici sono gli stessi delle porte TTL, in modo da semplificare al massimo la "migrazione" di un sistema da logiche standard a PLD.

I primi PLD che studieremo consentono di realizzare semplici funzioni combinatorie. Vedremo quindi come è possibile estenderne le funzionalità utilizzando delle opportune macrocelle di uscita. Passeremo quindi all'analisi dei circuiti logici programmabili più avanzati (*Complex PLD* o *CPLD*) per accennare infine alla struttura interna delle matrici di porte programmabili (*Field-Programmable Gate Array* o *FPGA*).

12.2 Matrici Logiche Programmabili (PLA)

I dispositivi logici programmabili più semplici prendono il nome di PLA, acronimo di (*Programmable Logic Array*), e sono stati introdotti nel Capitolo 10. Una PLA è in grado di realizzare funzioni logiche combinatorie, espresse come somma di prodotti. I parametri che caratterizzano una PLA sono: il numero di ingressi N , il numero di termini prodotto P , il numero di uscite M . Il numero di termini prodotto disponibili in PLA commerciali è generalmente compreso fra 4 e 16, indipendentemente dal numero di ingressi, in modo da limitare il numero di interconnessioni programmabili e la complessità del piano AND. Abbiamo visto che una PLA è costituita da un insieme di porte AND e di porte OR opportunamente interconnesse. Per semplificare la realizzazione circuitale di una PLA è opportuno trasformare preventivamente le funzioni logiche in forma NAND-NAND oppure NOR-NOR. Nel caso di PLA realizzate in tecnologia bipolare si utilizzano due piani di tipo NAND. Entrambi vengono realizzati mediante delle matrici di diodi (che realizzano funzioni di tipo AND) con invertitori in uscita. Per programmare una PLA di tipo bipolare è necessario selezionare, mediante opportuni segnali di ingresso, i fusibili da disconnettere e applicare quindi un'opportuna tensione di programmazione (10-30 V) ai loro capi. In questo modo i collegamenti metallici dei fusibili selezionati si interrompono, programmando la PLA. Ovviamente, l'operazione di programmazione di una PLA bipolare è irreversibile: il circuito è programmabile una sola volta. Le PLA in tecnologia MOS utilizzano due piani di tipo NOR, realizzati in logica NMOS o pseudo-NMOS. Per programmare la PLA vengono spesso utilizzati dispositivi MOS con doppia gate, una delle quali isolata (gate "fluttuante" o floating-gate). Questa tecnologia, mutuata da quella delle memorie EPROM ed EEPROM, consente di riprogrammare più volte la PLA. Si rimanda al capitolo sulle memorie per una disamina delle modalità di programmazione e del principio di funzionamento dei dispositivi floating-gate.

12.3 Dispositivi PAL

Gli svantaggi principali delle prime PLA disponibili commercialmente erano gli elevati costi di produzione e i notevoli tempi di propagazione. Queste limitazioni sono dovute alla presenza di due piani AND e OR entrambi programmabili. La presenza di matrici programmabili, infatti, richiede una elevata occupazione di area, una tecnologia più avanzata e comporta sensibili ritardi di propagazione. Per ovviare a questi inconvenienti sono stati introdotti i dispositivi PAL (*Programmable Array Logic*). Men-

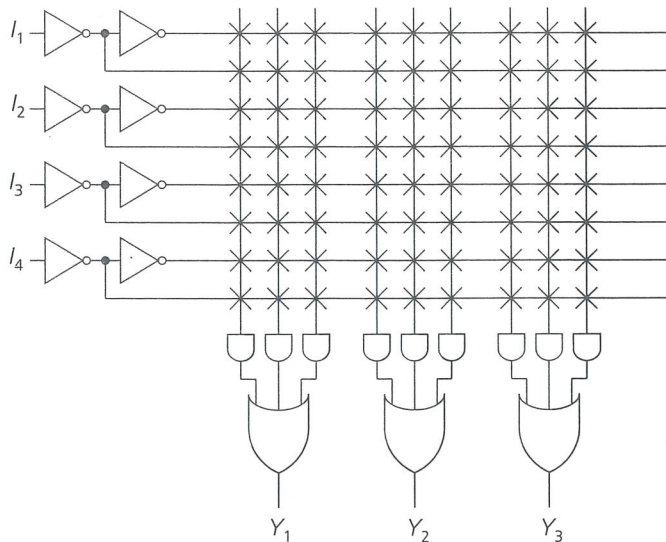


Figura 12.1 Schema compatto di una semplice PAL.

tre in una PLA sono programmabili sia il piano AND che il piano OR, una PAL è caratterizzata dall'aver il piano OR fisso, per cui il solo piano AND è programmabile. La Figura 12.1 mostra un esempio di PAL. Per il piano AND si è utilizzata la notazione compatta introdotta nel Paragrafo 10.9. Nella semplice PAL di Figura 12.1 si hanno 4 ingressi e 3 uscite mentre il numero di termini prodotto per ogni uscita è pari a 3.

Le uscite del piano AND non possono essere condivise da più porte OR. Se un termine prodotto è necessario per due uscite deve essere calcolato due volte, come mostra la Figura 12.2 che riporta la PAL programmata per realizzare le funzioni indicate:

$$\begin{aligned}
 Y_1 &= I_1 I_2 + I_3 \bar{I}_4 + I_2 \bar{I}_3 \\
 Y_2 &= I_1 \bar{I}_2 I_3 + I_1 I_4 \\
 Y_3 &= I_2 \bar{I}_3 + I_1 I_4
 \end{aligned}
 \tag{12.1}$$

Si noti che per le uscite Y_2 e Y_3 un termine prodotto è collegato sia alla variabile I_1 che a quella negata, per cui l'uscita della corrispondente porta AND è sempre 0. Co-

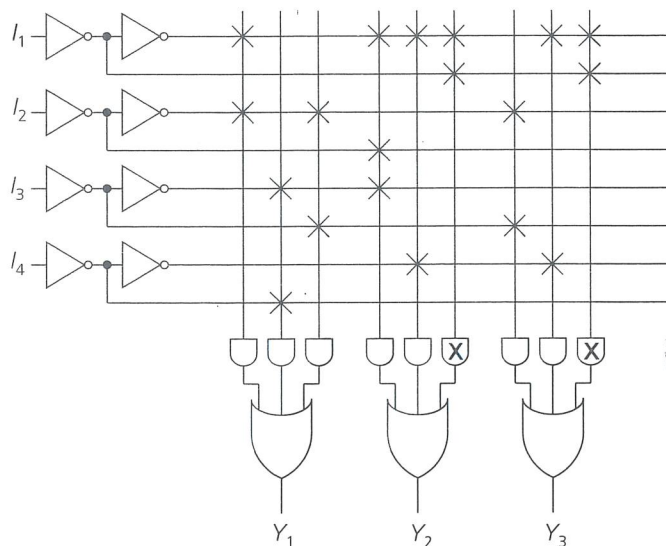


Figura 12.2 PAL programmata per realizzare le funzioni (12.1).

tuni siste-
matica la
vello. Gli
entari fino
tri a scor-
lelle porte
la logiche

ombinato-
o delle op-
program-
ara interna
'GA).

ronimo di
PLA è in
prodotti. I
vero di ter-
ponibili in
dal numero
i e la com-
me di por-
realizzazio-
oni logiche
tecnologia
ti mediante
ri in uscita.
ante oppor-
'opportuna
ollegamenti
vviamente,
cuito è pro-
iani di tipo
.A vengono
(gate "flut-
rie EPROM
apitolo sul-
ipio di fun-

rano gli ele-
azioni sono
presenza di
na tecnolo-
a questi in-
logic). Men-

me evidenzia la Figura 12.2, una AND programmata per avere uscita sempre 0 viene evidenziata con una X sullo schema circuitale.

12.3.1 Logica in due passi

Nelle PAL commerciali, il numero di termini prodotto disponibili per ogni uscita è compreso fra 8 e 16. Per consentire la realizzazione di funzioni logiche che richiedono un numero maggiore di mintermini si utilizza una logica in due passi. Come mostra la Figura 12.3, alcune delle uscite della PAL (in questo caso la sola Y_3) vengono nuovamente inviate come ingressi del piano AND; ciò consente di realizzare funzioni più complesse al prezzo di un maggior ritardo di propagazione.

Consideriamo, per esempio, il caso in cui la PAL di Figura 12.3 debba essere utilizzata per realizzare le due funzioni seguenti:

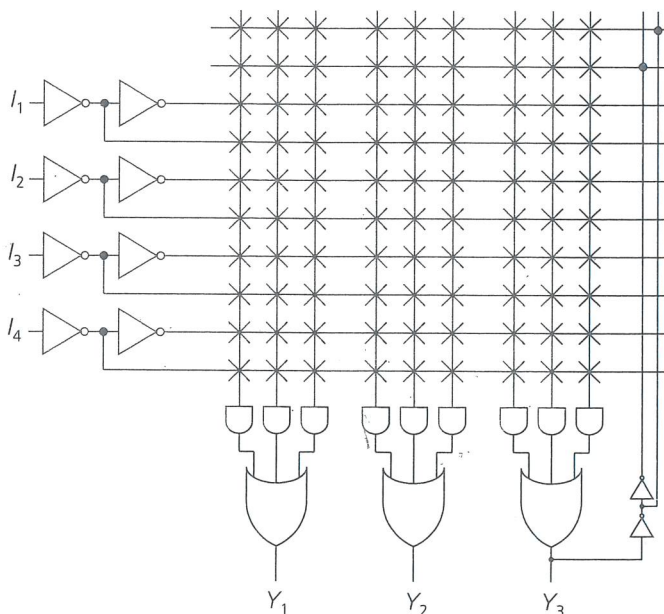
$$\begin{aligned} Y_1 &= I_1 I_2 I_3 + I_1 \bar{I}_2 \bar{I}_3 + \bar{I}_1 I_2 \bar{I}_3 + \bar{I}_1 \bar{I}_2 I_3 \\ Y_2 &= I_1 I_2 I_3 + \bar{I}_2 \bar{I}_3 I_4 + I_1 I_4 \end{aligned} \quad (12.2)$$

L'uscita Y_2 richiede tre termini prodotto e rientra nei limiti della PAL, mentre per l'uscita Y_1 il numero di termini prodotto è pari a 5, ed è pertanto maggiore dei termini prodotto disponibili in questa PAL per ogni uscita. Dobbiamo pertanto riscrivere l'espressione per l'uscita Y_1 , utilizzando come termine di "aiuto" (*helper term*) l'uscita Y_3 :

$$\begin{aligned} Y_3 &= I_1 I_2 I_3 + I_1 \bar{I}_2 \bar{I}_3 \\ Y_1 &= Y_3 + \bar{I}_1 I_2 \bar{I}_3 + \bar{I}_1 \bar{I}_2 I_3 \end{aligned} \quad (12.3)$$

La corrispondente mappa di programmazione è mostrata nella Figura 12.4. La logica a due passi aumenta la flessibilità di una PAL, a prezzo di un maggior ritardo di propagazione. Per la PAL di Figura 12.4, il ritardo per l'uscita Y_1 è all'incirca il doppio del ritardo relativo all'uscita Y_2 .

Figura 12.3 Schema di una PAL che consente di realizzare una logica in due passi.



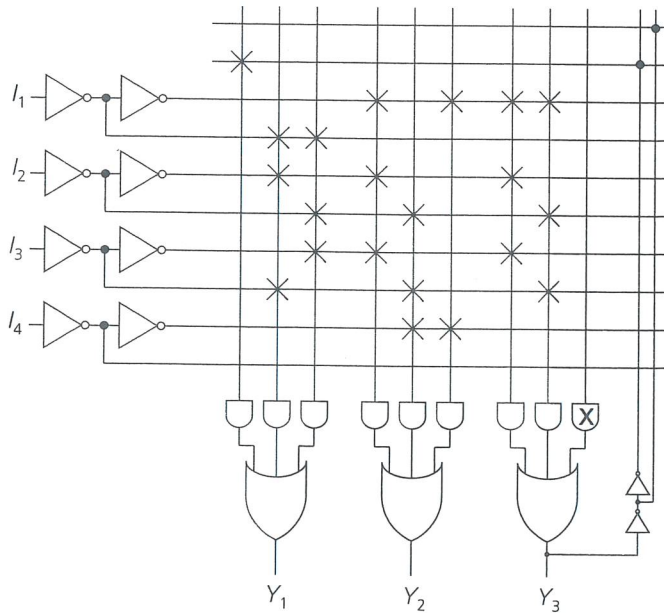


Figura 12.4 Programmazione della PAL di Figura 12.3.

12.3.2 Programmazione della polarità in uscita

Supponiamo di dover utilizzare la PAL di Figura 12.3 per realizzare le due funzioni seguenti:

$$Y_1 = I_1 I_2 I_3 + I_1 \bar{I}_2 \bar{I}_3 + \bar{I}_1 I_2 \bar{I}_3 + \bar{I}_1 \bar{I}_2 I_3 \quad (12.4)$$

$$Y_2 = I_1 + I_2 + I_3 + I_4$$

Rispetto all'esempio del paragrafo precedente, il numero di termini prodotto per l'uscita Y_2 è pari a 4, ed è pertanto maggiore dei termini prodotto disponibili nella nostra PAL. Inoltre, non è possibile utilizzare l'uscita Y_3 come termine di "aiuto", dato che Y_3 è necessaria al calcolo di Y_1 . In definitiva, le due funzioni non possono essere realizzate con la PAL di Figura 12.3. D'altro canto, se dovessimo calcolare \bar{Y}_2 invece di Y_2 il problema sarebbe risolto, visto che \bar{Y}_2 richiede la valutazione di un solo termine prodotto:

$$\bar{Y}_2 = \bar{I}_1 \bar{I}_2 \bar{I}_3 \bar{I}_4$$

Molti PLD dispongono della possibilità di poter programmare la polarità dell'uscita, come mostra la Figura 12.5. Ogni uscita è provvista di una porta XOR. Un ingresso della XOR è collegato all'uscita del piano OR. L'altro ingresso della XOR è collegato a livello logico alto tramite un resistore di pull-up, ma può essere portato a massa in fase di programmazione del dispositivo, tramite un'opportuna interconnessione programmabile.

In questo modo è possibile invertire l'uscita della PAL. Infatti, indicando con Z l'uscita del piano OR e con Y l'uscita della XOR, quando il secondo ingresso della XOR è a zero risulta:

$$Y = Z \text{ XOR } 0 = Z \quad (12.5)$$

mentre se il secondo ingresso della XOR è programmato in modo da rimanere a livello logico alto si ha:

$$Y = Z \text{ XOR } 1 = \bar{Z} \quad (12.6)$$

La Figura 12.6 mostra la mappa di programmazione della PAL, per realizzare le due funzioni Y_1 e Y_2 assegnate.

Figura 12.5 PAL con polarità dell'uscita programmabile.

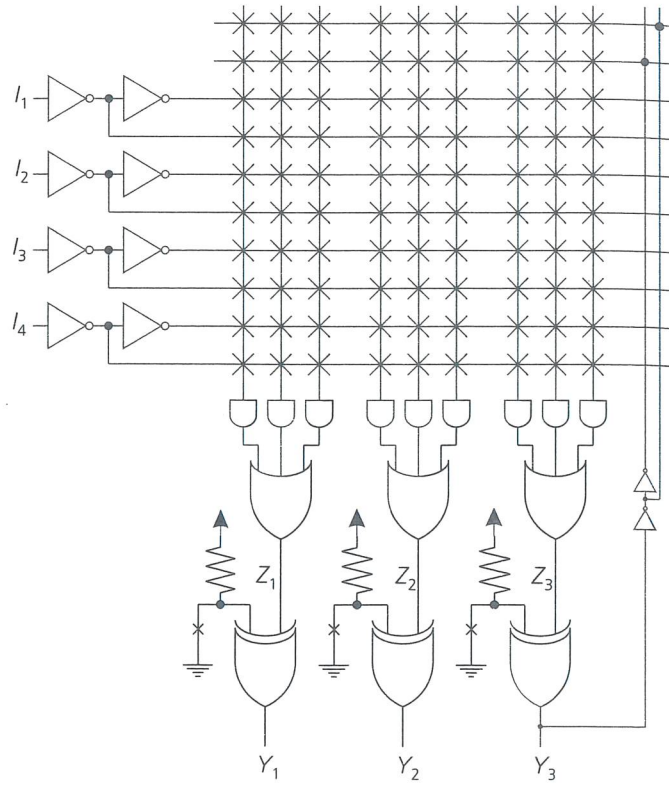
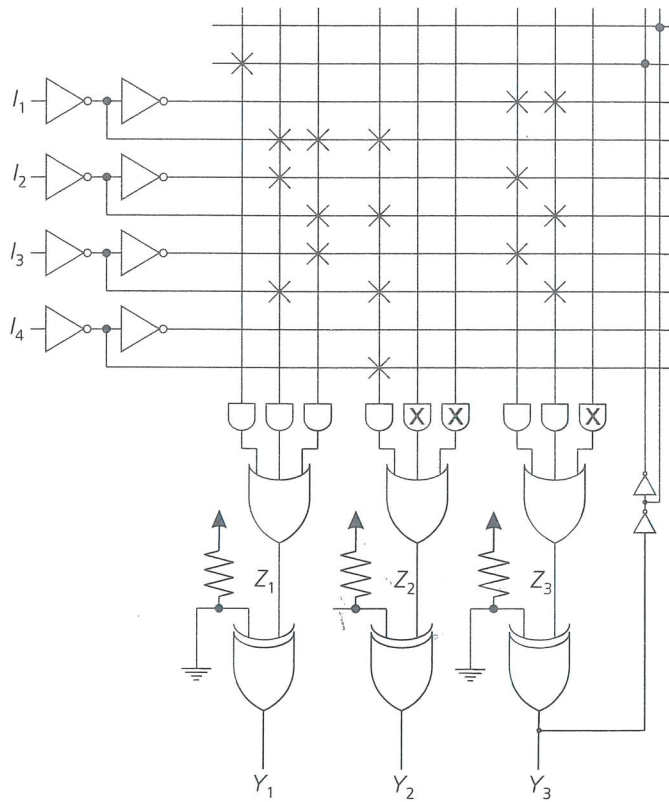


Figura 12.6 Programmazione della PAL di Figura 12.5.



12.3.3 Terminali di Ingresso/Uscita programmabili

La Figura 12.7 mostra la PAL di Figura 12.5 in cui l'uscita Y_3 è utilizzata come terminale di ingresso/uscita programmabile (*I/O pin*). Dopo la porta XOR, che realizza l'inversione programmabile, è presente un buffer di uscita tristate. Il segnale di abilitazione della porta tristate è pilotato da una uscita dedicata del piano AND; se quest'uscita è 0 il buffer si porta in condizioni di alta impedenza.

Un pin di I/O può essere utilizzato in tre modi diversi:

- Se il circuito è programmato in modo tale che il segnale di abilitazione del buffer tristate sia sempre 0 (Figura 12.8), il pin corrispondente può essere utilizzato come ingresso, sfruttando la coppia di invertitori necessaria per realizzare la logica a due passi.
- Se gli ingressi della porta AND sono mantenuti a livello logico alto da circuiti di pull-up, il segnale di abilitazione della porta tristate è alto, e il pin corrispondente può essere utilizzato come uscita oppure come *helper* per realizzare una logica a due passi. Questo caso è indicato nella Figura 12.9, dove tutti gli ingressi della porta AND che pilota il segnale di abilitazione del buffer tristate sono disconnessi (non sono mostrate le reti di pull-up).
- il segnale di abilitazione del buffer tristate dipende dagli ingressi del PLD. In questo caso il pin può essere dinamicamente utilizzato come terminale di ingresso o di uscita. Una tipica applicazione è quella di un'interfaccia collegata a un bus condiviso con altre periferiche.

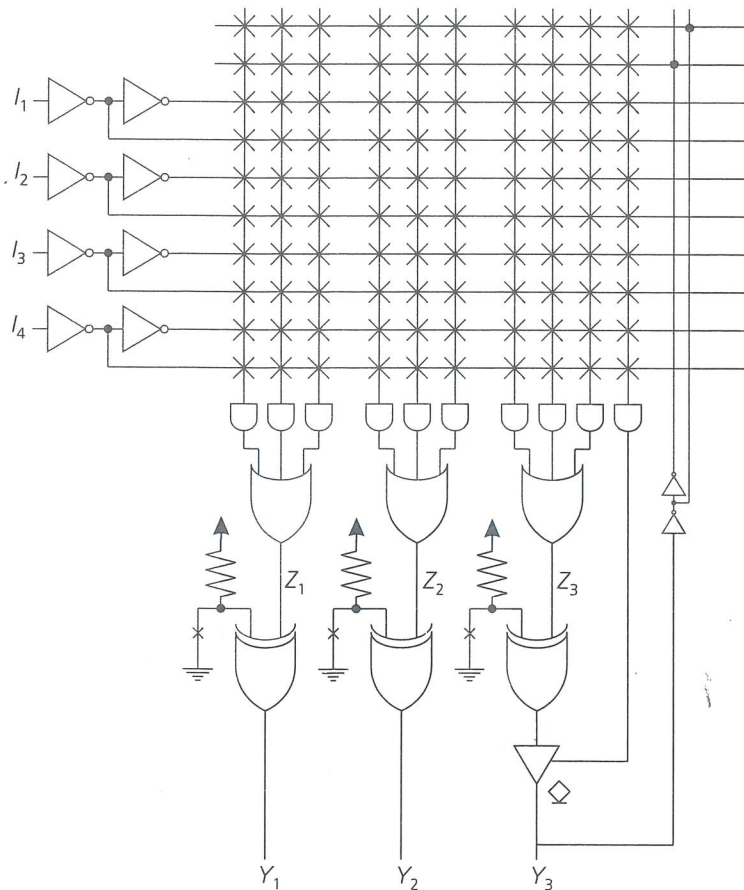
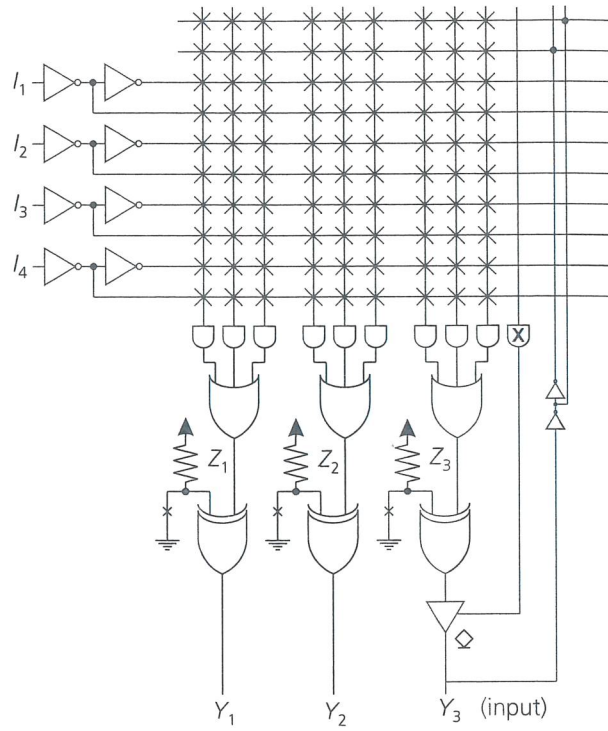


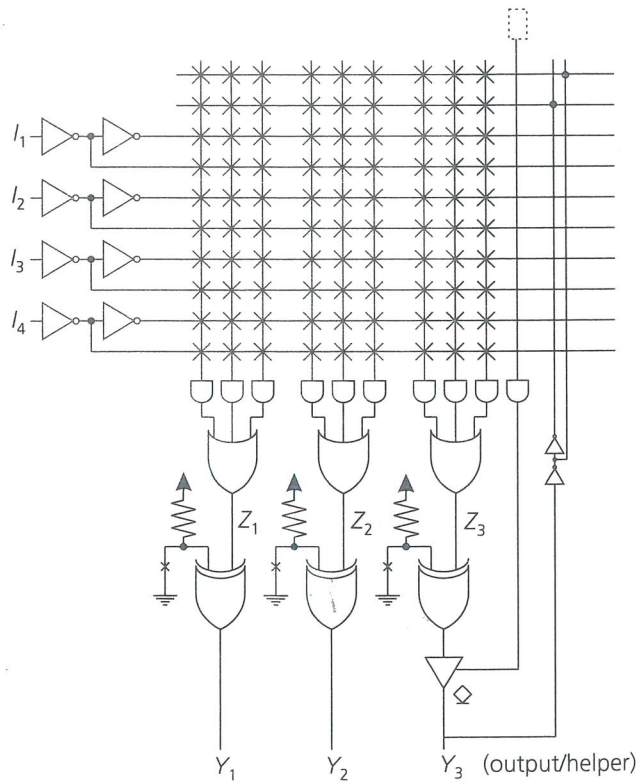
Figura 12.7 PAL con I/O programmabile.

Figura 12.8 Utilizzo di un pin di I/O programmabile come ingresso.



La disponibilità di terminali di I/O programmabili aumenta notevolmente la flessibilità di un PLD. Ad esempio una PLA con 10 terminali di I/O programmabili può essere

Figura 12.9 Utilizzo di un pin di I/O programmabile come uscita o come termine di helper per logiche in due passi.



utilizzata, a seconda delle esigenze, per realizzare sei funzioni logiche di quattro ingressi, oppure quattro funzioni logiche di quattro ingressi con due termini di helper oppure tre funzioni logiche di quattro ingressi con tre termini di helper ecc.

12.4 PLD sequenziali

I circuiti logici programmabili sequenziali includono al loro interno dei flip-flop, in modo da poter realizzare sistemi sequenziali come contatori, registri a scorrimento ecc. Un esempio di PAL sequenziale è riportato nella Figura 12.10. In corrispondenza di Y_2 e Y_3 sono inseriti due flip-flop. Le uscite dei due flip-flop, oltre che essere disponibili come terminali di uscita della PAL, sono inviate come feedback all'interno del piano AND, in modo da consentire la realizzazione di macchine a stati finiti. Si noti che il segnale di clock è comune a tutti i flip-flop, come dovrebbe essere in ogni sistema sincrono ben progettato.

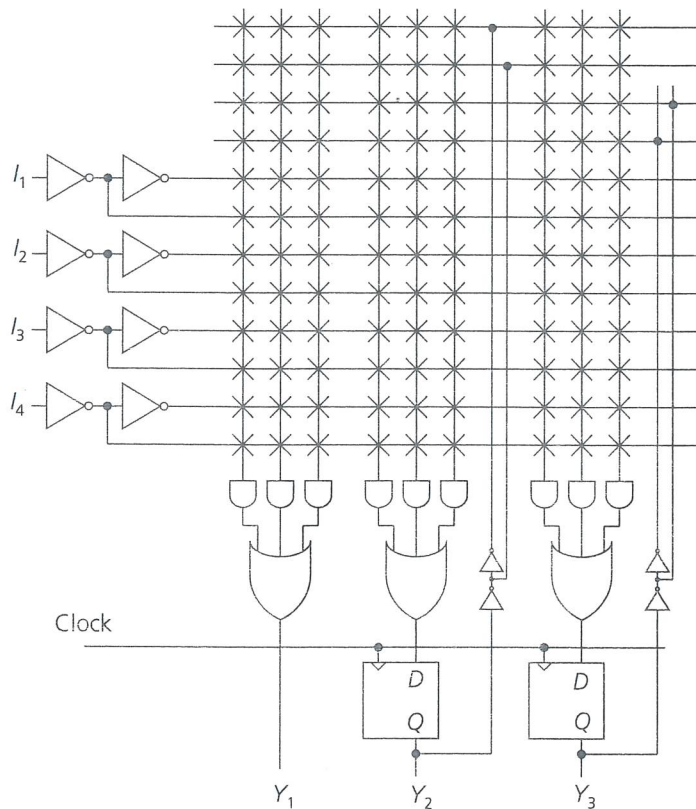
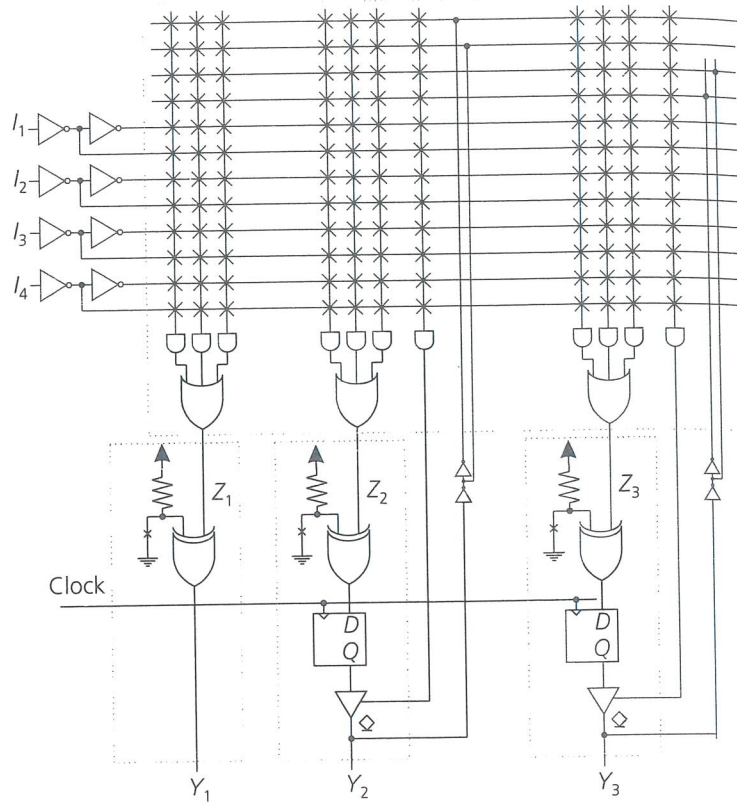


Figura 12.10 PAL sequenziale.

12.4.1 Macrocelle di uscita

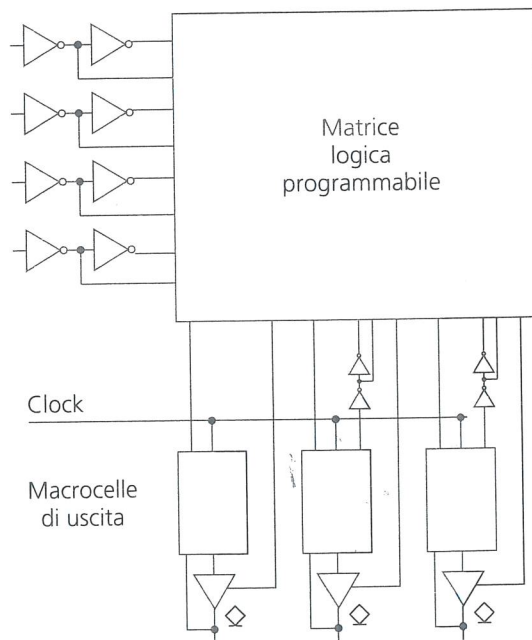
Tutti gli accorgimenti visti nei paragrafi precedenti per aumentare la flessibilità delle PAL combinatorie possono essere utilizzati anche nelle PAL sequenziali. La Figura 12.11 mostra l'introduzione dell'inversione programmabile dell'uscita e dei terminali di I/O programmabili per la PAL della Figura 12.10.

Figura 12.11 PAL sequenziale con macrocella di uscita.



La Figura 12.12 riporta un diagramma a blocchi della PAL di Figura 12.11. Da questa figura si evidenzia come una PAL possa in generale essere suddivisa in:

Figura 12.12 Schema a blocchi semplificato della PAL di Figura 12.11.



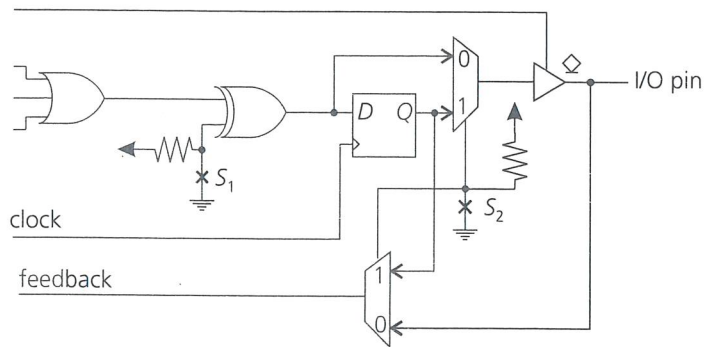


Figura 12.13 Macrocella di uscita di una PAL.

- Una matrice logica programmabile, costituita da un piano AND programmabile e da un piano OR fisso. Gli ingressi della matrice programmabile sono costituiti sia da ingressi primari del circuito che da eventuali termini di feedback.
- Un insieme di macrocelle di uscita, anch'esse programmabili.

Nell'esempio di Figura 12.11 le macrocelle di uscita consentono di invertire la polarità dell'uscita e di controllare i terminali di I/O programmabili. Peraltro, a causa dell'introduzione dei registri di uscita, non è possibile realizzare una logica in due passi. Inoltre non è possibile realizzare una macchina a stati le cui uscite possano essere poste in una condizione di alta impedenza.

Per risolvere questi inconvenienti è sufficiente modificare le macrocelle di uscita. La Figura 12.13 mostra lo schema di una macrocella più generale di quella di Figura 12.11. Due multiplexer, comandati dalla connessione programmabile S_2 , consentono di selezionare la modalità di funzionamento della macrocella.

In modalità "sequenziale" (si veda la Figura 12.14) il segnale di feedback corrisponde all'uscita Q del flip-flop. Si noti che in questa modalità operativa l'uscita della macrocella può essere posta in condizione di alta impedenza, grazie alla presenza del

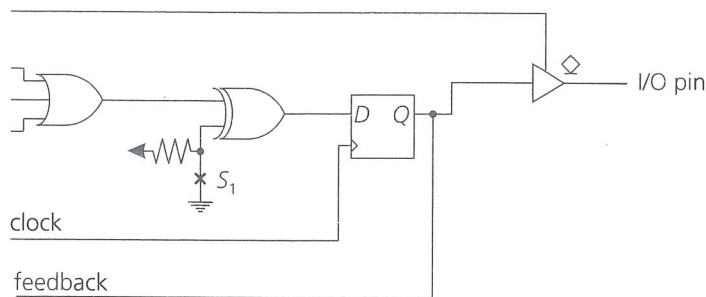


Figura 12.14 Macrocella di uscita in modalità "sequenziale".

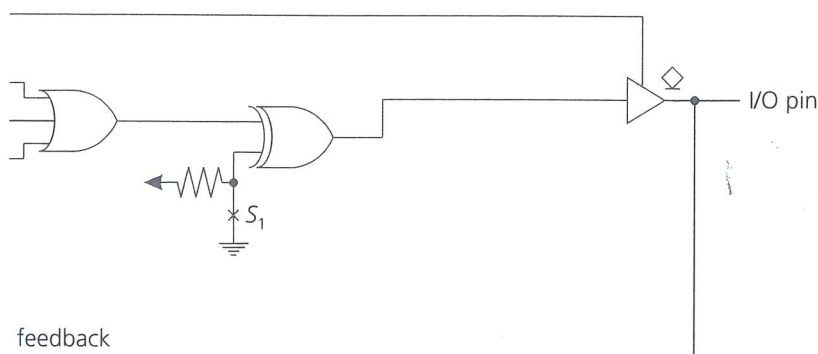


Figura 12.15 Macrocella di uscita in modalità "combinatoria".

buffer tristate. In modalità "combinatoria" (si veda la Figura 12.15) l'uscita della porta XOR (che realizza l'inversione programmabile di polarità) viene inviata direttamente al buffer di uscita. Il segnale di feedback corrisponde con l'uscita del buffer, in modo da poter realizzare una logica a due passi oppure utilizzare il pin come terminale di ingresso della PAL.

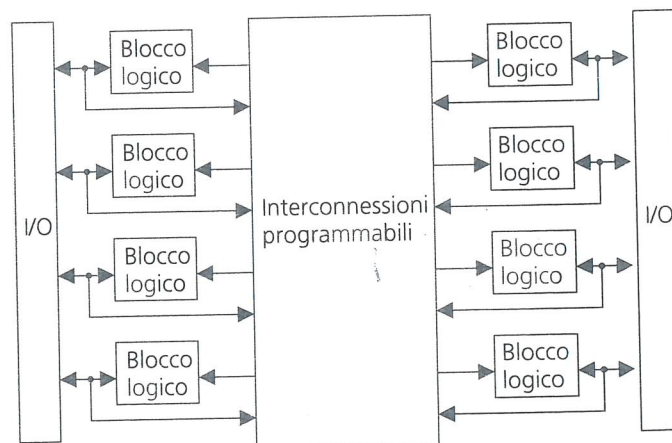
12.5 PLD complessi (CPLD)

Il progressivo sviluppo della tecnologia CMOS ha consentito la realizzazione di circuiti programmabili di complessità sempre maggiore. Un parametro utilizzato molto di frequente per stabilire la capacità di una PLD è il numero di porte equivalenti. Diremo che un dato PLD ha una capacità di 1000 porte equivalenti se consente di realizzare delle funzioni logiche per le quali sarebbero necessarie 1000 porte NAND a due ingressi. La definizione, per quanto non rigorosa, è utile in quanto, nelle fasi iniziali di un progetto, consente di valutare in prima approssimazione quanti PLD sono necessari per realizzare un dato sistema. Per esempio, un dispositivo PAL commerciale come la PAL22V10 con 22 ingressi per la matrice AND programmabile e 10 uscite, ha una capacità dell'ordine di 800 porte equivalenti. Nel caso dei circuiti di tipo PAL, aumentare la capacità corrisponde ad aumentare il numero di ingressi del piano AND programmabile. Ciò comporta un aumento insostenibile del fan-in delle porte che compongono il piano AND, con significativi peggioramenti in termini di tempi di propagazione. Per esempio, nella PAL22V10 il piano AND è costituito da 132 porte a 44 ingressi.

Invece di aumentare il numero di ingressi del piano AND è più conveniente aumentare la capacità di un dispositivo logico programmabile integrando più blocchi logici, ognuno simile a una PAL, su di uno stesso chip. I vari blocchi logici sono collegati fra loro mediante opportune interconnessioni programmabili, come mostra lo schema semplificato di Figura 12.16. Si parla, in questo caso di PLD Complesse o CPLD.

Le architetture delle varie CPLD attualmente disponibili, se possono tutte riportarsi allo schema di principio mostrato nella Figura 12.26, differiscono sia per la struttura interna dei blocchi logici che della matrice di interconnessione. Per quanto riguarda i blocchi logici essi sono sempre costituiti da una matrice AND programmabile e da un insieme di macrocelle di uscita. Come nelle PAL studiate nei paragrafi precedenti, ogni macrocella di un CPLD può essere programmata in modo da invertire eventual-

Figura 12.16 Schema di principio di una CPLD.



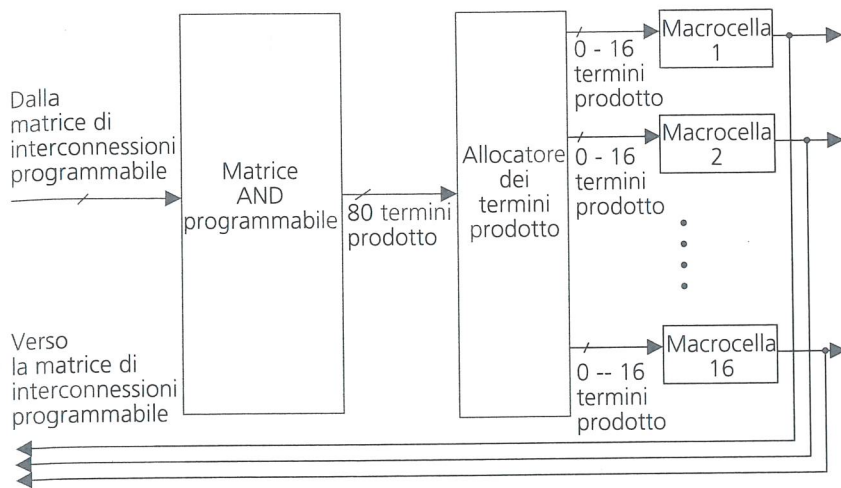


Figura 12.17 Schema semplificato del blocco logico delle CPLD della famiglia Cypress 370.

mente il segnale di uscita. Inoltre, ogni macrocella, includendo un flip-flop, può essere configurata come “combinatoria” o “sequenziale”. L’uscita della macrocella può essere inviata nuovamente in ingresso alla matrice AND in modo da realizzare un feedback locale. In un CPLD è inoltre possibile inviare l’uscita di una macrocella, attraverso il sistema di interconnessioni programmabili, all’ingresso del piano AND di un’altra macrocella, realizzando in questo modo un feedback globale. Alcuni blocchi logici includono delle macrocelle nascoste (*buried macrocells*), le cui uscite vengono utilizzate esclusivamente come feedback e non per pilotare i terminali di I/O.

I blocchi logici delle CPLD più avanzate consentono la allocazione variabile e la condivisione dei termini prodotto. A titolo esemplificativo, nella Figura 12.17 si riporta uno schema di principio del blocco logico delle CPLD della famiglia 370 della Cypress. L’uscita del piano AND, costituita da 80 termini prodotto, è inviata a un blocco programmabile di allocazione che smista per ognuna delle 16 macrocelle di uscita, alcuni termini prodotto, in numero variabile da 0 a 16. In questo modo, alle macrocelle dedicate al calcolo di funzioni logiche “semplici” verranno inviati pochi termini prodotto, consentendo alle altre macrocelle il calcolo di funzioni logiche più complesse. Inoltre, uno stesso termine prodotto può essere condiviso fra più macrocelle, come accadeva nelle PLA.

Negli ultimi anni, la capacità dei CPLD è aumentata in modo esponenziale, passando da poche migliaia di porte equivalenti fino a valori superiori a 10^5 porte equivalenti, con un numero di macrocelle variabile da alcune decine fino ad alcune centinaia. A titolo di esempio, la Tabella 12.1 riporta le caratteristiche salienti delle CPLD della famiglia FLEX10K della Altera.

Tabella 12.1 Caratteristiche dei CPLD della famiglia Altera FLEX10K.

Caratteristiche	EPF10K10	EPF10K30	EPF10K50	EPF10K100
Capacità	10.000	30.000	50.000	100.000
Numero blocchi logici	576	1728	2888	49.992
Numero flip-flop	720	1968	3184	5392
Numero pin I/O	150	246	310	406

Fortunatamente, accanto allo sviluppo dei CPLD (e degli FPGA che studieremo fra breve) si è assistito a un corrispondente sviluppo dei programmi di progettazione assistita al calcolatore (CAD). In questo modo, il progettista può concentrarsi sulla defini-

zione ad alto livello del sistema da sviluppare, lasciando a opportuni programmi di sintesi, ottimizzazione e fitting il compito di sfruttare al meglio le potenzialità dei dispositivi programmabili.

12.6 Matrici di porte programmabili (FPGA)

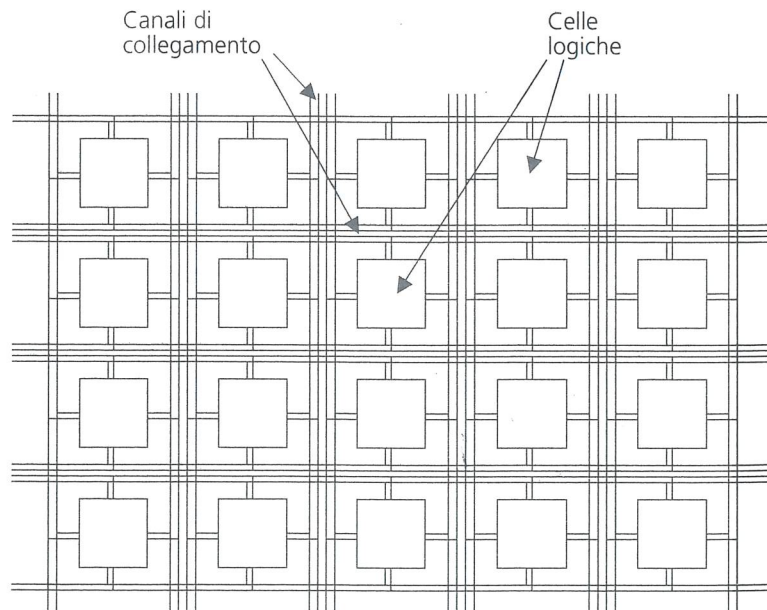
Un primo dispositivo gate-array è il gate-array programmabile a livello di maschera, che è costituito da una matrice regolare di celle logiche tutte uguali fra loro, integrate su di uno stesso chip che prende il nome di master. Le varie celle logiche di un master non sono collegate fra di loro. Come mostra la Figura 12.18, ogni singolo master viene "personalizzato" o programmato realizzando le sole linee di metal (localizzate in opportuni canali di collegamento orizzontali e verticali) che interconnettono le varie celle logiche in modo da realizzare la funzione desiderata.

Rispetto alla progettazione di un intero circuito integrato, l'utilizzo di gate-array programmabili a livello maschera consente sia di ridurre i costi (in quanto i master sono tutti uguali fra loro e vengono quindi progettati una volta per tutte e prodotti in larga scala), sia di ridurre i tempi di sviluppo (in quanto per personalizzare un master è necessario realizzare le sole interconnessioni metalliche, mentre tutti gli altri passi di processo, quali la realizzazione delle diffusioni, dei transistori ecc., sono già stati realizzati in precedenza).

Questi vantaggi sono bilanciati da un peggiore utilizzo dell'area di Silicio e da un aumento dei tempi di propagazione. Entrambi questi svantaggi sono dovuti al fatto che le celle logiche non sono ottimizzate per nessuna specifica applicazione e che, inoltre, le interconnessioni sono molto più lunghe rispetto a quelle che si avrebbero in un circuito realizzato ad hoc, con un sensibile aumento delle capacità parassite.

Le matrici di porte programmabili "sul campo" (*Field-Programmable Gate Array*, FPGA) rappresentano uno sviluppo della tecnologia dei gate-array programmabili a livello maschera. In un FPGA sono realizzate nel chip di silicio non solo la matrice di porte logiche ma anche le interconnessioni. A differenza di un gate-array programmato a livello maschera, le interconnessioni sono programmabili, utilizzando tec-

Figura 12.18 Schema di principio di una matrice di porte (gate-array).



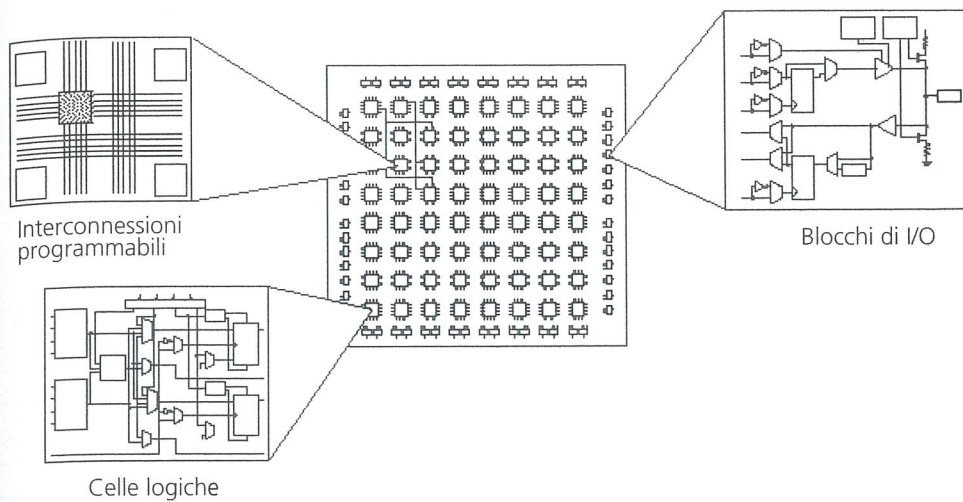


Figura 12.19 Struttura interna di un FPGA.

nologie del tutto analoghe a quelle che abbiamo studiato nei paragrafi precedenti. La Figura 12.19 mostra la struttura interna di un FPGA ed evidenzia la presenza di opportune celle di ingresso/uscita programmabili.

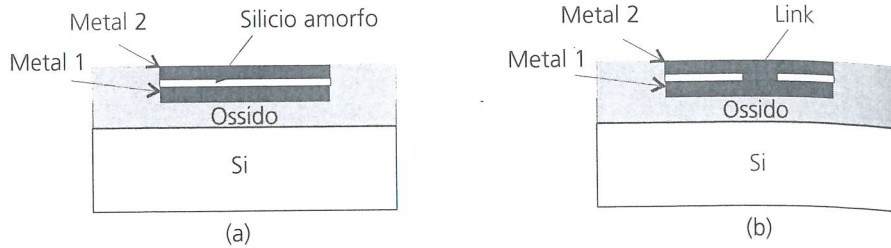
Da un punto di vista applicativo, molte caratteristiche degli FPGA e dei CPLD sono simili fra loro. In effetti, se confrontiamo le Figure 12.16 e 12.19 vediamo che sia i CPLD che gli FPGA sono costituiti da interconnessioni programmabili e da opportune celle o blocchi logici.

La principale differenza è legata al fatto che le celle logiche di un FPGA hanno in generale una funzionalità ridotta rispetto alla combinazione di piano AND programmabile e macrocella di un CPLD. D'altro canto il numero di celle logiche di un FPGA è molto maggiore rispetto al numero di blocchi logici di un CPLD. In sintesi, un FPGA ha un'architettura "a grana fine" rispetto all'architettura "a grana grossa" di un CPLD.

Questa differenza comporta alcune differenze nelle caratteristiche di CPLD e FPGA. Per esempio, la presenza di un elevato numero di celle logiche consente di rendere disponibili un numero maggiore di flip-flop in un FPGA rispetto a un CPLD.

Un'altra differenza riguarda i tempi di propagazione. La struttura "a grana fine" di un FPGA consente di realizzare funzioni estremamente complesse collegando opportunamente in cascata varie celle logiche, ognuna delle quali è costituita da un numero limitato di porte logiche e quindi ha un ritardo di propagazione ridotto. In definitiva, il ritardo di propagazione complessivo di un FPGA è determinato in larga misura dal numero di livelli logici necessari per realizzare una assegnata funzione e dai ritardi introdotti dalle interconnessioni programmabili presenti sul chip. Ne consegue che il ritardo di propagazione complessivo di un FPGA è molto variabile a seconda del particolare sistema da realizzare, e può essere calcolato solo al termine di tutte le fasi di sviluppo del progetto. In un CPLD, invece, le funzioni vengono realizzate mediante logica a due passi (feedback locale) o, al più, mediante un feedback globale (un collegamento fra l'uscita di una macrocella e l'ingresso al piano AND di un'altra macrocella). Il ritardo di un blocco logico di una CPLD è peraltro maggiore rispetto a quello di una cella logica di un FPGA. Pertanto, il ritardo introdotto da una CPLD può essere valutato facilmente nelle fasi iniziali dello sviluppo di un progetto ed è poco dipendente dalla complessità del progetto stesso.

Figura 12.20 Programmazione con antifusibile: (a) prima della programmazione; (b) dopo la programmazione.



12.6.1 Tecniche di programmazione

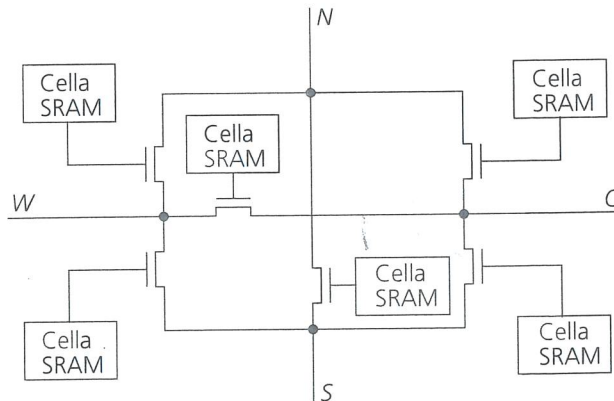
Mentre nelle PAL e nei CPLD, come per le PLA, la tecnica di programmazione è basata sull'utilizzo di transistori a gate flottante (che verranno discussi trattando delle memorie ROM programmabili), nelle FPGA si adoperano antifusibili (nel caso di circuiti programmabili una sola volta) oppure memorie RAM statiche, dette SRAM (nel caso di FPGA riprogrammabili).

La Figura 12.20 mostra lo schema di un collegamento programmabile mediante antifusibile. Dalla Figura si nota che i contatti fra due livelli di metallo (in questo caso particolare, il primo e il secondo livello di metallo) comprendono un sottile strato di silicio amorfo, non conduttore. L'applicazione di una opportuna differenza di potenziale ai capi del contatto consente di rompere il sottile strato isolante, realizzando un contatto elettrico a bassa resistenza fra le due linee di interconnessione in metal1 e in metal2.

La Figura 12.21 mostra l'utilizzo di celle di memoria SRAM per realizzare delle interconnessioni programmabili. Ad ogni cella di memoria è collegata la gate di un transistore NMOS, che opera come porta di trasmissione. Utilizzando sei locazioni di memoria e altrettante porte di trasmissione nel circuito della Figura 12.21 si possono effettuare tutti i possibili collegamenti fra i quattro terminali *N*, *W*, *E* e *S*. Il vantaggio di disporre di un sistema riprogrammabile viene pagato con una maggiore occupazione di area e con maggiori ritardi di propagazione rispetto alla tecnologia basata su antifusibili.

Le memorie SRAM sono inoltre volatili, per cui la programmazione viene "persa" dopo aver scollegato l'alimentazione alla FPGA. In questi circuiti la mappa dei collegamenti viene in pratica memorizzata in una piccola ROM, esterna alla FPGA. Nei primi istanti dopo l'applicazione della tensione di alimentazione, il contenuto della ROM viene trasferito nella SRAM interna programmando così l'FPGA.

Figura 12.21 Utilizzo di celle di memoria SRAM e porte di trasmissione NMOS per realizzare interconnessioni programmabili.



La Tabella 12.2 riassume le caratteristiche delle differenti tecniche di programmazione utilizzate nei dispositivi logici programmabili.

Tabella 12.2 Caratteristiche delle differenti tecniche di programmazione utilizzate nei dispositivi logici programmabili.

Programmazione	Tecnologia	Riprogrammabile	Volatile	Dispositivo
Fusibile	bipolare	no	no	PAL
EEPROM	CMOS	si	no	PAL, CPLD
MOS Floating gate	CMOS	no	no	FPGA
SRAM	CMOS	si	si	FPGA

Come accennato in precedenza, è spesso possibile utilizzare indifferentemente un FPGA o un CPLD per realizzare una stessa applicazione. In alcuni casi si utilizza la sigla FPGA per riferirsi indifferentemente all'una o all'altra classe di dispositivi programmabili. In Tabella 12.3 sono riportate come esempio le caratteristiche salienti degli FPGA della famiglia XC4000 della XILINX.

Tabella 12.3 Caratteristiche degli FPGA della famiglia XILINX XC4000.

Caratteristiche	XC4003	XC4010	XC4025	XC4085
Capacità (gate equiv.)	3000	10.000	25.000	85.000
Matrice blocchi logici	10 × 10	20 × 20	32 × 32	56 × 56
Numero di flip-flop	360	1120	2560	7168
Numero di pin I/O	80	160	256	448

Riferimenti bibliografici

- D. PELLERIN, M. HOLLEY, *Practical design using programmable logic*, Prentice-Hall Int., 1991.
- J.F. WAKERLY, *Digital Design, Principles and Practices*, 2° ed., Prentice-Hall Int., 1994.
- K. SKAHILL, *VHDL for programmable logic*, Addison-Wesley, 1996.
- M.J.S. SMITH, *Application-specific integrated circuits*, Addison-Wesley, 1997.

13.1 Introduzione

La realizzazione dei sistemi digitali può essere sviluppata con diverse metodologie di progetto e con differenti implementazioni. La via più seguita fino a una decina di anni fa è stata quella di implementare i sistemi utilizzando i componenti logici standard SSI e MSI presentati nella prima parte di questo testo, componenti che vengono opportunamente assemblati su schede con circuiti stampati per realizzare le funzioni desiderate.

Un differente approccio alla realizzazione dei sistemi è quello basato sull'uso di componenti logici programmabili, come i CPLD e gli FPGA introdotti nel Capitolo 12, con i quali, sfruttando le crescenti potenzialità di questi componenti sia in termini di complessità di funzioni logiche realizzabili che in termini di capacità, ossia di numero di porte equivalenti, è possibile realizzare sottosistemi e sistemi di notevole complessità con pochi di questi componenti. D'altra parte l'elevata flessibilità richiesta per questi componenti e la possibilità di programmazione delle funzioni hanno un costo, sia in termini di area di silicio utilizzata, che in termini di prestazioni dinamiche (tempi di propagazione e velocità di funzionamento), rispetto a soluzioni che prevedano la realizzazione diretta delle funzioni desiderate sul silicio senza far uso di componenti programmabili. Tale costo è giustificato nel caso di realizzazioni di prototipi, o anche per produzioni di piccola scala per le quali l'elevato costo di progettazione di sistemi "dedicati" non può essere ammortizzato dalla produzione di un elevato numero di componenti di costo più ridotto. Un altro campo di applicazione dei componenti suddetti è legato alla necessità di poter disporre di una sezione di logica programmabile in un sistema più ampio; infine può essere molto utile la possibilità di poter riprogettare parti del sistema modificando semplicemente il software di programmazione nei componenti programmabili.

Per ottimizzare le prestazioni dei componenti e ridurre il costo della realizzazione dei circuiti è tuttavia più efficace e vantaggiosa la realizzazione degli interi sottosistemi digitali in circuiti a larghissima scala di integrazione (VLSI), che permettono una progettazione specifica dei vari circuiti con significativi vantaggi per quanto riguarda la compattazione dell'area utilizzata, del consumo di potenza e delle prestazioni dinamiche dei circuiti stessi.

Un approccio ampiamente utilizzato è quello del progetto a celle standard (*Standard Cell Design*). In questa metodologia di progetto il sistema da implementare nel chip di silicio viene realizzato utilizzando una libreria di celle standard, contenute in apposite librerie. Le celle standard contengono circuiti di tipo SSI o MSI, già ottimizzati per le specifiche funzioni richieste, e standardizzati in termini di lay-out, in modo da poter essere semplicemente interconnessi e piazzati sul chip. In altri termini il posizionamento e le interconnessioni sul chip delle differenti celle standard della libreria è equivalente a ciò che precedentemente era realizzato sulla piastra stampata, posizio-

nando e collegando tra di loro i circuiti integrati dei componenti standard. Una possibile libreria di celle standard è riportata nella Tabella 13.1.

Tabella 13.1 Libreria di celle standard SSI e MSI.

Tipo di cella	Caratteristiche
Inverter/buffer/tristate	
Porta NAND/AND	2-8 ingressi
Porta NOR/OR	2-8 ingressi
Porta XOR/XNOR	2-3 ingressi
Porta AOI/OAI	
Multiplexer	2-8 ingressi
Decoder	2-8 ingressi
Schmitt trigger	invertente, non invertente
Adder	normale, veloce
Bistabile	D, asincrono, sincrono
Registro	D, JK, asincrono, sincrono
Circuiti I/O	In, out, tristate, bidirezionali
ROM	

Il progetto di sistemi digitali nei quali ogni singolo transistor viene ottimizzato per la funzione da realizzare è detto progetto Full Custom (*Full Custom Design*). In questo caso possono essere utilizzate strutture non convenzionali e modalità di sincronizzazione particolari per ottenere le prestazioni migliori in termini di occupazione di area o di velocità di funzionamento. Questa metodologia è la più onerosa in termini di tempo (e costo) di progettazione, per contro è quella che permette la massima compattazione di area a parità di funzioni realizzate e le più spinte prestazioni dinamiche. La metodologia di progetto Full Custom viene quindi utilizzata per componenti VLSI per i quali è prevista una produzione in larga scala, come memorie RAM e ROM, microprocessori, FPGA, o per componenti con sezioni analogiche come i convertitori A/D, D/A e gli anelli ad aggancio di fase (*Phase-Locked Loops, PLL*).

Nei circuiti a larga scala di integrazione la tecnologia dominante è quella CMOS, per le sue caratteristiche di trascurabile dissipazione di potenza statica e di facile integrazione, anche se per alcuni dei blocchi logici (circuiti I/O, circuiti di clock, buffer) può essere vantaggioso ricorrere anche ad altre tecnologie, come quelle bipolari o BiCMOS. In questo capitolo verranno introdotte le principali strutture, basate su tecnologia CMOS, che sono utilizzate nei sistemi VLSI per migliorare le prestazioni dinamiche dei circuiti e per ridurre significativamente l'occupazione di area a parità di funzioni realizzate. Queste sono le logiche FCMOS, le logiche a porte di trasmissione, le logiche dinamiche.

13.2 Logiche complesse FCMOS

Come si è detto, la tecnologia CMOS è quella più utilizzata nei circuiti VLSI. Con questa tecnologia è possibile realizzare, oltre alle porte base NAND o NOR, anche porte logiche che realizzano direttamente funzioni logiche più complesse di quelle elementari. Lo schema generico di una porta CMOS che realizza una funzione complessa